MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2019/2020

ECP4136 – JAVA TECHNOLOGY

(RE / TE)

17 OCTOBER 2019
2:30 p.m - 4:30 p.m
(2 Hours) – OPEN BOOK

INSTRUCTIONS TO STUDENT

1. This Question paper consists of 7 pages including cover page with 3 Questions only.

2. Attempt **ALL** questions. The distribution of the marks for each question is given.

3. This is an Open-Book examination.

4. Please write all your answers in the Answer Booklet provided.

Instructions:

Figure 1 shows the UML diagram for an offline quiz application, consisting a list of Multiple-Choice Questions (MCQ), and each MCQ has 4 options. The UML diagram will be referred for Question 1, 2 and 3.

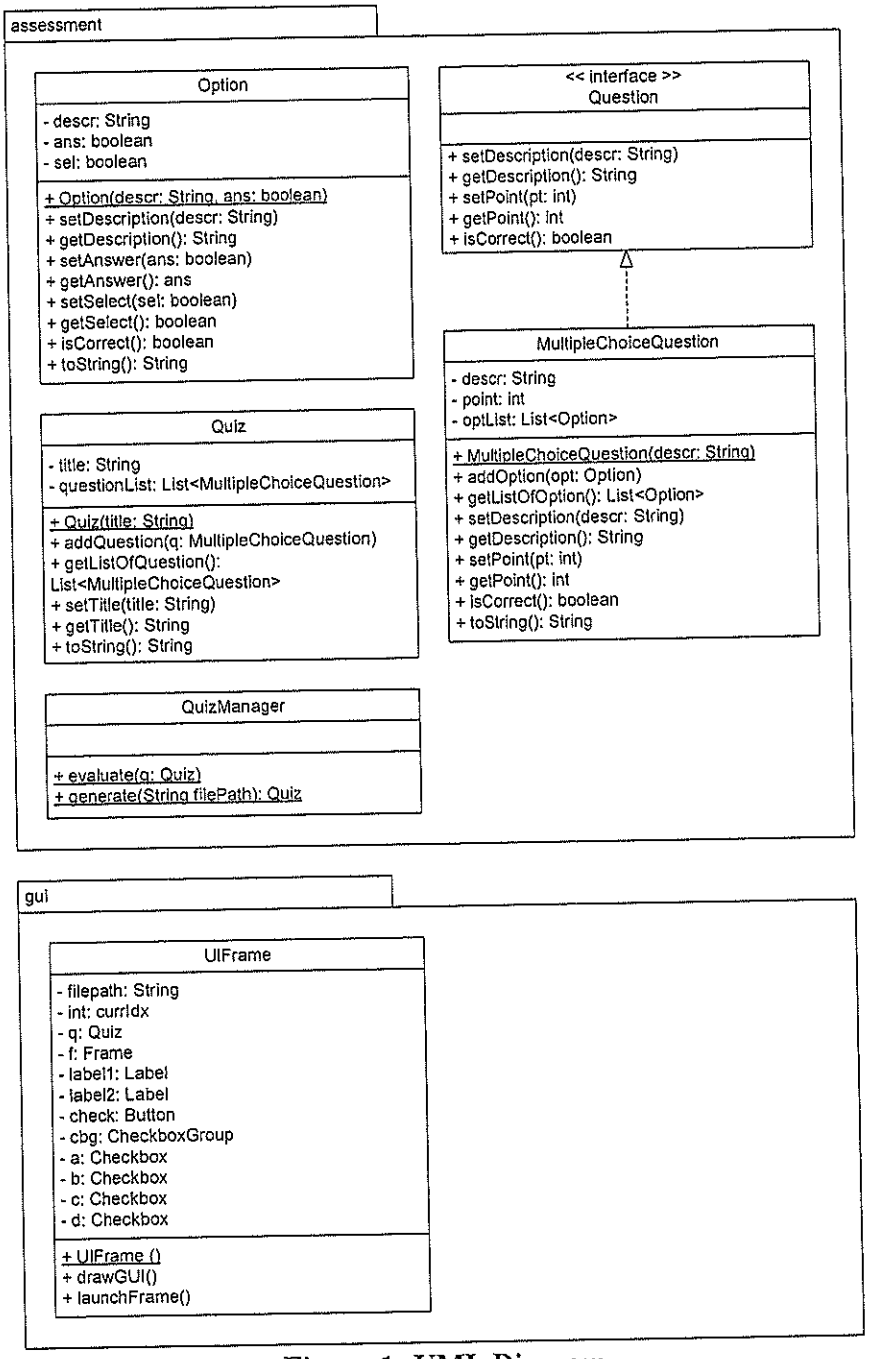

Figure 1: UML Diagram

Continued .....

Question 1

a) Implement the `Option` class according to the following requirements:

- `descr` attribute is the description of the option; `ans` attribute tells whether the option is an answer for the question; `sel` attribute tells whether the user has selected the option.
- The constructor saves the description and answer in the corresponding attributes. It also initializes `sel` attribute to `false`.
- `setDescription` method and `getDescription` method are the setter and getter for the `descr` attribute, respectively.
- `setAnswer` method and `getAnswer` method are the setter and getter for the `ans` attribute, respectively.
- `setSel` method and `getSel` method are the setter and getter for the `sel` attribute, respectively.
- `isCorrect` method checks whether the user has selected the option correctly, by matching `ans` attribute and `sel` attribute.
- `toString` method returns the `descr` attribute, with the inclusion of square brackets as shown in Figure 2.

```
[ ] 105
```

Figure 2

[1+2+1+1+1+1+1 marks]

b) Create the Question interface.

[3 marks]

Continued .....

c) Implement the **`MultipleChoiceQuestion`** class according to the following requirements:

- **`descr`** attribute is the description of the question; **`point`** attribute is the point awarded for answering the question correctly; **`optList`** attribute is used to keep the list of **`Option`** objects.
- The constructor saves the description in the corresponding attribute. It also initializes the **`point`** attribute to **`1`** and creates an **`ArrayList`** object for the **`optList`** attribute.
- **`setDescription`** method and **`getDescription`** method are the setter and getter for the **`descr`** attribute, respectively.
- **`setPoint`** method and **`getPoint`** method are the setter and getter for the **`point`** attribute, respectively.
- **`addOption`** method adds an **`Option`** object into the **`optList`**.
- **`getListOfOption`** method is the getter for **`optList`**.
- **`isCorrect`** method checks whether the user has answered the question correctly, by examining every **`Option`** object in the **`optList`**.
- **`toString`** method organizes the descriptions of the question and options according to the example shown in Figure 3.

```
What is 67 + 33 - 5
[ ] 105
[ ] 93
[ ] 64
[ ] 95
```

Figure 3

[1+2+1+1+1+1+1+1 marks]

Continued .....

Question 2

a) Implement the `Quiz` class according to the following requirements:
- `title` attribute is the quiz title; `questionList` attribute is used to keep the list of `MultipleChoiceQuestion` objects.
- The constructor saves the title in the corresponding attribute. It also creates an `ArrayList` object for the `questionList` attribute.
- `setTitle` method and `getTitle` method are the setter and getter for the `title` attribute, respectively.
- `addQuestion` method adds a `MultipleChoiceQuestion` object into the `questionList`.
- `getListOfQuestion` method is the getter for `questionList`.

[1+1+1+1+1 marks]

b) Implement the `QuizManager` class.
- The `generate` method returns a `Quiz` object after performing the following tasks.
  - Exception will be printed out if it happens.

[2 marks]

  - Create a `File` object according to the `filePath` argument, which indicates the location of a text file that consists of all questions of a quiz. Figure 4 shows the content of a sample text file.

[1 marks]

  - Read the file line-by-line. The first 5 letters of each line form a command, followed by the information used to prepare the `Quiz`, `MultipleChoiceQuestion`, and `Option` objects. The commands are described in Table 1.

[4 marks]

- The `evaluate` method computes the total points scored by the user, by examining all questions in the quiz.

[3 marks]

Continued .....

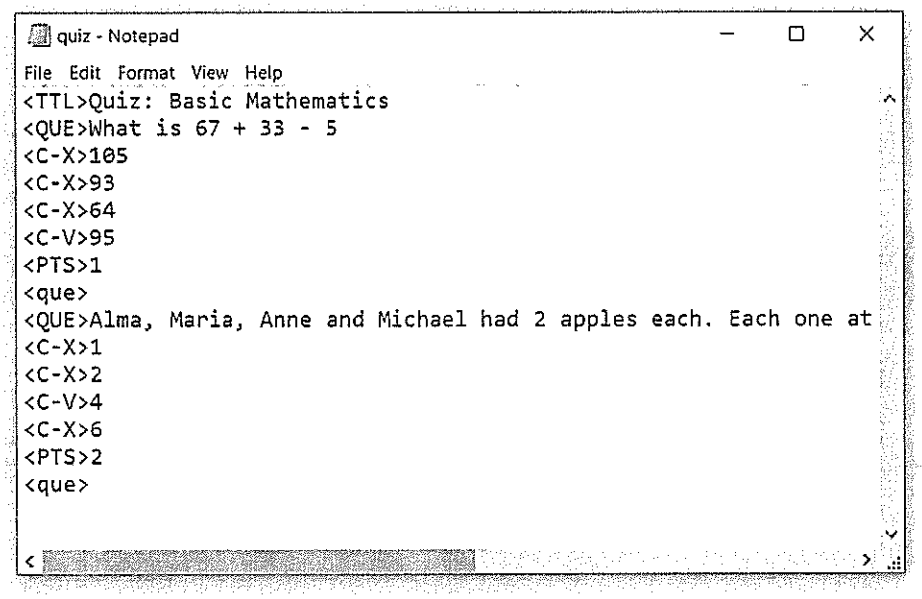

Figure 4

Table 1

Command	Description
`<TTL>`	The command is followed by the quiz title. A new **`Quiz`** object will be created.
`<QUE>`	The command is followed by the description of a question. A new **`MultipleChoiceQuestion`** object will be created.
`<que>`	**`<QUE>`** and **`<que>`** are used to enclose the options and point belongs to the question. The **`MultipleChoiceQuestion`** object will be added to the **`Quiz`**.
`<C-X>`	Followed by the description of an option. The option **IS NOT** the answer. **`Option`** object will be added to the **`MultipleChoiceQuestion`**.
`<C-V>`	Followed by the description of an option. The option **IS** the answer. **`Option`** object will be added to the **`MultipleChoiceQuestion`**.
`<PTS>`	Followed by the points awarded for the question.

Continued .....

Question 3

Implement the **`UIFrame`** class that can be launched to present the Graphical User Interface (GUI) as shown Figure 5.

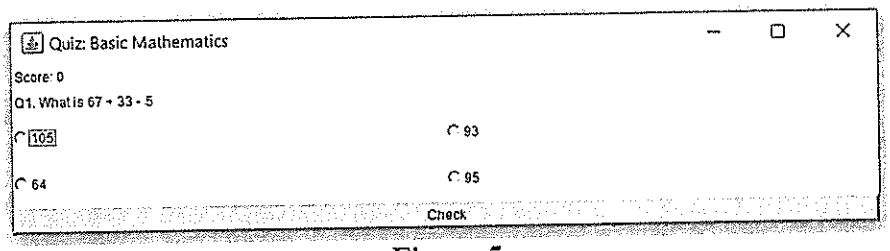

Figure 5

- **`filePath`** attribute is the file path of the text file for a quiz. **`currIdx`** attribute is the index of the question shown on the user interface. **`q`** attribute refers to the **`Quiz`** object. The other attributes are the GUI components used to form the user interface. You may use more components for your program.

[4 marks]

- The constructor initializes the **`currIdx`** attribute to `0` and generates the **`Quiz`** object according to the **`filePath`** attribute. Assume **`filePath`** is `"C:/project/quiz.txt"`.

[2 marks]

- **`launchFrame`** method prepares the GUI. Event handler for the *Check* button is added to the button.

[3 marks]

- By clicking the *Check* button, **`Checkbox`** objects will be examined to determine the option chosen by the user. **`currIdx`** will be updated to the subsequent question. **`drawGUI`** method will be invoked.

[3 marks]

- **`drawGUI`** method updates the total score and question descriptions shown in the GUI, according to **`currIdx`** attribute. If **`currIdx`** attribute is more than or equal to the number of questions, only total score will be updated.

[3 marks]

End of Page